

Understanding about Automotive security and application in DCV project

Instructor: luan1.nguyen@lge.com

Agenda

- ❖ Introduce about security for automotive today
- ❖ ASDL - Automotive security development life cycle
- ❖ Integration/System Verification for Security Feature
- ❖ Vulnerability Assessment
- ❖ Security standards (framework, protocol/mechanism, algorithm,...)

Introduce

- ❖ Today, have more and more automotive use a operating system (OS) for control/support driver and provide entertainment services.
=> It create many opportunities and challenger for security automotive.
- ❖ The insecure automotive system is a REAL problem (not just as potential, or not exist as many people believe)



Introduce

Refer link:

<https://tinhte.vn/thread/hacker-co-the-hack-xe-hoi-dieu-khien-xe-tat-bat-may-xe-va-lam-nhieu-tro-khac-tu-xa.2486702/>

<https://plo.vn/xe-va-luat/lieu-xe-hoi-co-bi-chiem-quyen-dieu-khien-boi-hackers-469662.html>

<https://xedoisong.vn/cong-nghe/hack-va-dieu-khien-xe-bang-tin-nhan-dien-thoai-10237.html>

https://tinhte.vn/thread/hacker-co-the-hack-xe-hoi-dieu-khien-xe-tat-bat-may-xe-va-lam-nhieu-tro-khac-tu-xa.2486702/


ernet Explorer

Xe > Giao thông & PTVC khác

Hacker có thể hack xe hơi, điều khiển xe, tắt bật máy xe, và làm nhiều trò khác từ xa

Nam Air 5 năm • Bình luận: 32

Theo dõi Thông báo



Lời dẫn: Xe ô tô ngày càng trở nên thông minh, đó là điều có thật, chiếc xe từ một phương tiện vận chuyển, dần dần được thêm vào nhiều tiện nghi, bộ xử lý trung tâm cũng dần mạnh lên, và những chiếc xe tự lái không cần người thì đã và đang được thử nghiệm nhiều hơn, những chiếc xe thông minh "smart car" ngoài những tiện lợi của nó, thì kéo theo những nguy hiểm như: xe sẽ bị mất kiểm soát nếu bị hacker phá hoại, câu chuyện dưới đây do tạp chí Wired đăng tải, mình xin phép dịch nguyên văn gửi đến các bạn.

Trong 20 thư gửi hãng xe, có 16 hãng xe trả lời thư, hãng nào cũng xác nhận là xe của mình có một kết nối không dây nào đó, như wifi, bluetooth, radio, mạng 3G, 4G hoặc mạng di động, nhưng chỉ có 7 hãng xe nói có nhờ công ty bảo mật test xe, và chỉ 2 hãng xe nói có hệ thống quét hệ thống CAN để tìm mã độc.



Hai anh hacker này chỉ mới khai thác hệ thống Uconnect của Chrysler, ngoài ra còn có những hệ thống khác chưa được nghiên cứu như GM Onstar, Lexus Enform, Toyota Safety Connect, Hyundai Bluelink, và Infiniti Connection, nếu những hệ thống này tồn tại những lỗi tương tự, thì cũng dễ dàng bị khai thác một cách tương tự.

XE ĐỜI SỐNG www.xedoisong.vn

TIN TỨC XE MỚI THỬ XE THỊ TRƯỜNG CỎ

Thứ Hai 04/01/2021 16:39

CÔNG NGHỆ

Hack và điều khiển xe bằng tin nhắn điện thoại

Thư năm, 13/08/2015 13:38

Chia sẻ Thích 0 Chia sẻ

Chỉ bằng cách gửi các tin nhắn, mẫu xe thể thao Chevrolet Corvette ngoạn ngoạn nghe lời bật gạt nước và phanh dừng.



Chia sẻ

Các lo ngại về nguy cơ bị hack trộm, mở khóa cửa, chiếm quyền điều khiển xe bằng các thiết bị bên ngoài đang ngày càng trở thành mối lo ngại lớn của người dùng và của cả các nhà sản xuất ô tô. Hàng loạt các vụ hack thành công cho thấy ô tô ngày nay dễ bị tin tặc tấn công như thế nào.

Cách đây vài tuần, hai hacker người Mỹ đã tận dụng kẻ hở trên hệ thống Uconnect của Jeep để bẻ khóa đột nhập thành công chiếc Grand Cherokee. Trước đó, trong chương trình "60 Minutes", đài CBS cũng cho thấy một chiếc xe có thể trở thành nạn nhân bị tấn công từ xa dễ như thế nào. Những hệ thống kết nối như OnStar của GM, hay các loại chìa khóa điều khiển từ xa có thể là cầu nối lý tưởng để các hacker thực hiện một vụ tấn công.

Và mới đây, các hacker đến từ Đại học California lại hack thành công chiếc Chevrolet Corvette đời 2013 thông qua thiết bị ngoại vi kết nối với cổng chẩn đoán của xe. Sử dụng một phần mềm trên điện thoại, nhóm hacker này có thể ra lệnh cho xe bật gạt nước, đạp phanh và thực hiện nhiều can thiệp khác chỉ bằng cách gửi tin nhắn.

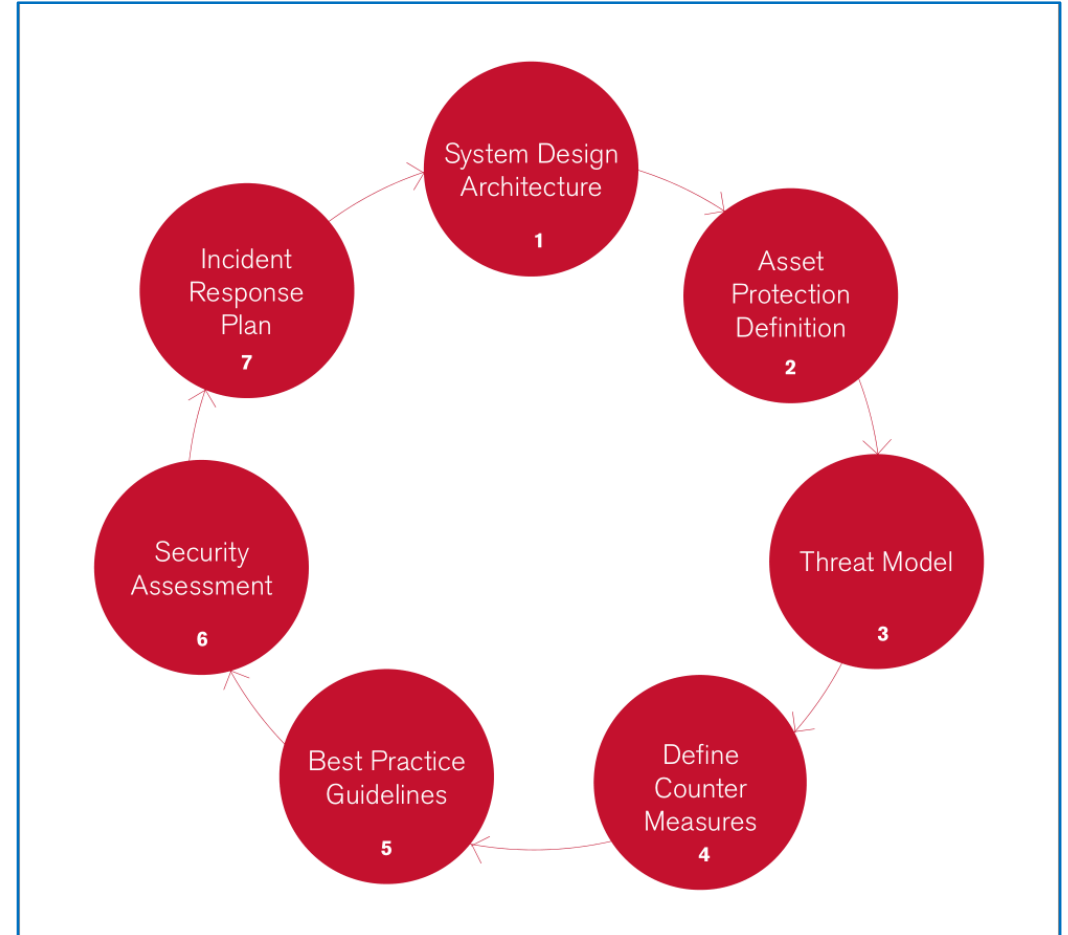
ASDL - Automotive secure development life cycle

❖ Main refer:

<https://www.nccgroup.com/globalassets/landing-pages/automotive/asdl-automotive-secure-development-lifecyclepdf/>

❖ File name: ASDL – Automotive Secure Development Lifecycle.pdf (16 pages)

- 1 Introduction
- 2 ASDL – The Automotive Secure Development Lifecycle
- 3 System Design/Architecture Review
- 4 Asset Protection Definition
- 5 Threat Modelling
 - 5.1 Spoofing Identity
 - 5.2 Tampering with Data
 - 5.3 Repudiation
 - 5.4 Information Disclosure
 - 5.5 Denial of Service
 - 5.6 Elevation of Privilege
- 6 Define Countermeasures
- 7 Best Practice Guidance
- 8 Security Assessment
- 9 Incident Response Plan



ASDL - Automotive security development life cycle (def)

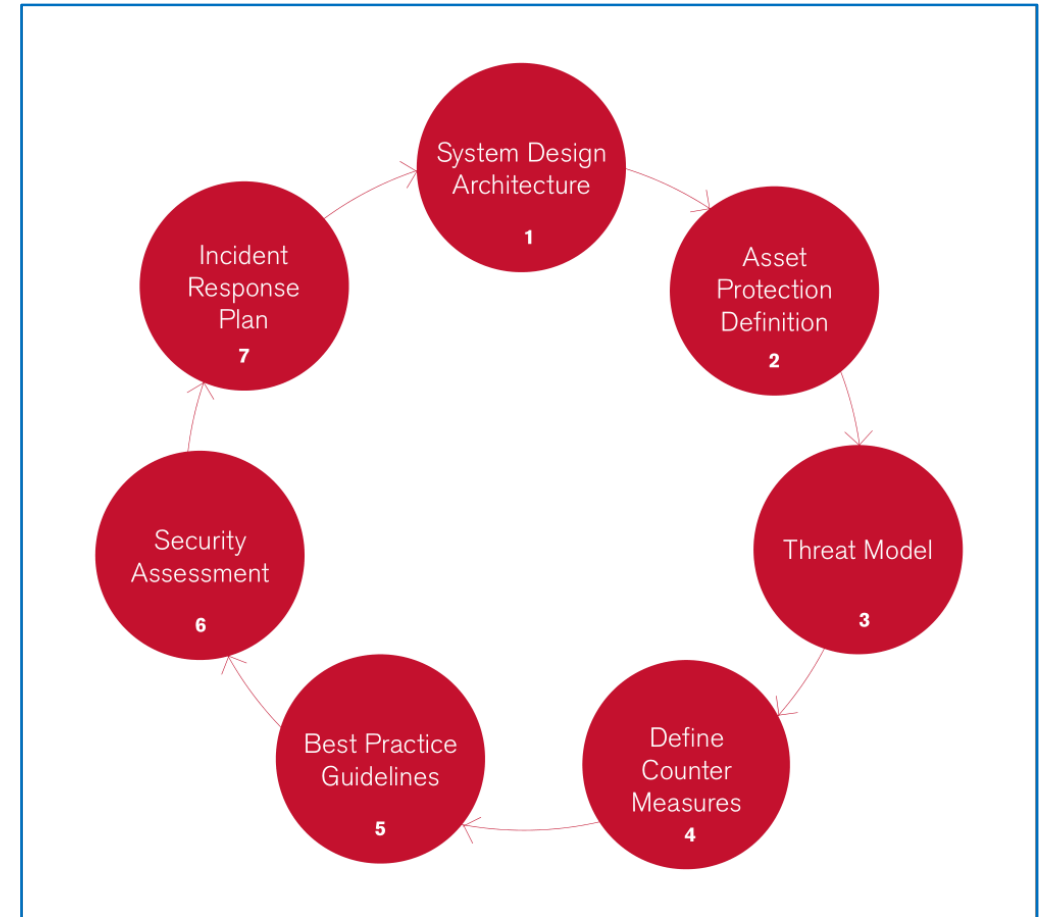
The Automotive Secure Development Lifecycle (ASDL) is a cost-effective hardware and software engineering approach to cyber-security assurance, covering the entire development lifecycle within the automotive world. The model is intended to provide security assurance at each stage in the development lifecycle of vehicles and vehicle components.

The ASDL consists of seven stages. Each stage is underpinned with training, and, as such, is seen as a core component of the model. The ASDL should be considered as a framework, rather than as a solution that replaces standards such as ISO 17799 or ISO 26262.

Refer:

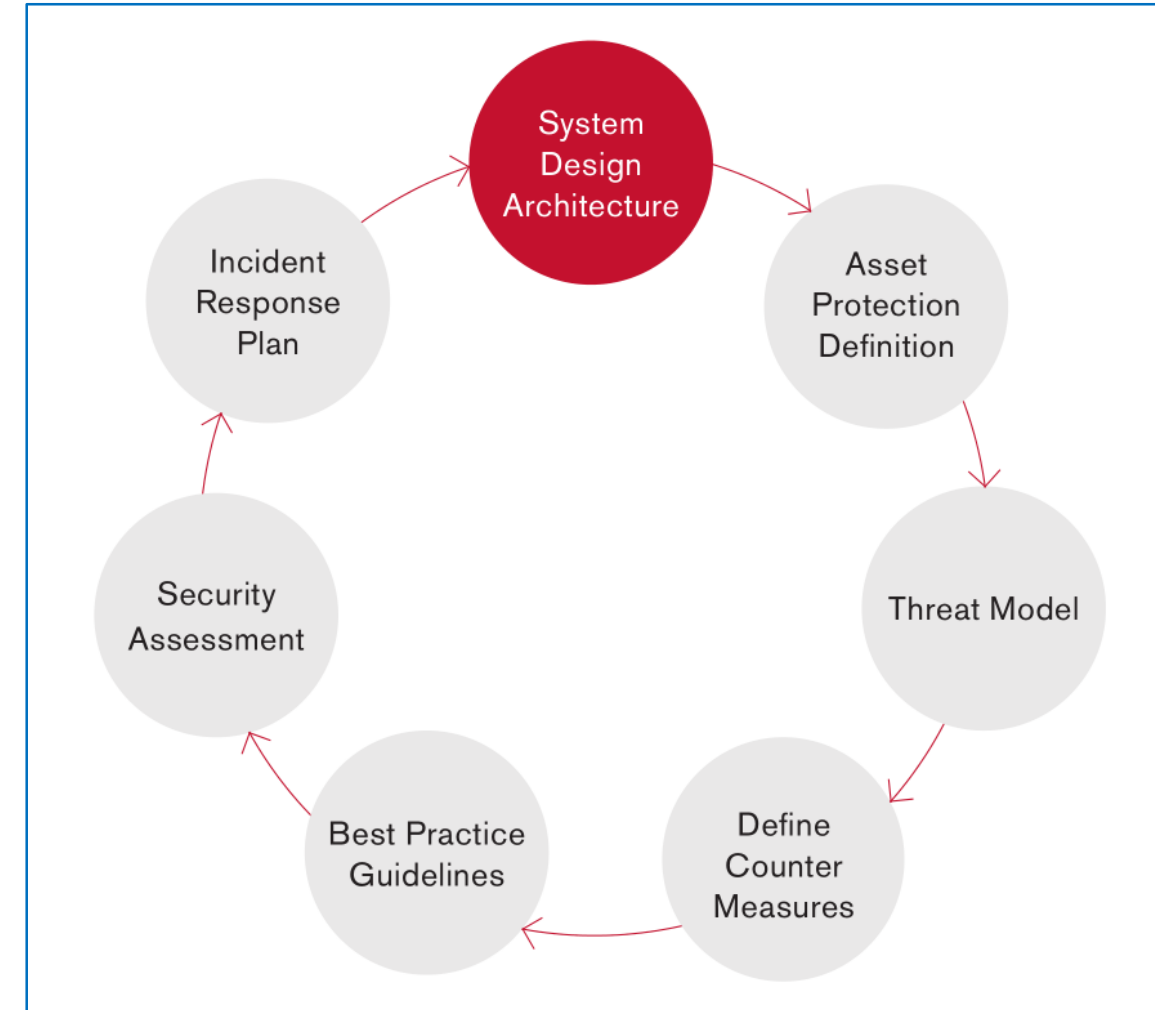
ISO 17799: <https://www.iso.org/standard/39612.html> (ISO/IEC 17799:2005 Information technology — Security techniques — Code of practice for information security management)

ISO 26262: <https://www.iso.org/standard/68383.html> (ISO 26262-1:2018 Road vehicles — Functional safety)



ASDL - System Design/Architecture

- Development of the requirement specification to help with the procurement or development of the right products or solutions
- Evaluation of the available security technologies, working closely with suppliers
- Advice on integration of security technologies and applications on diverse platforms
- Identification of process and technology improvements
- Advice on efficient use of security technologies and management of the associated risks



ASDL - Asset Protection Definition

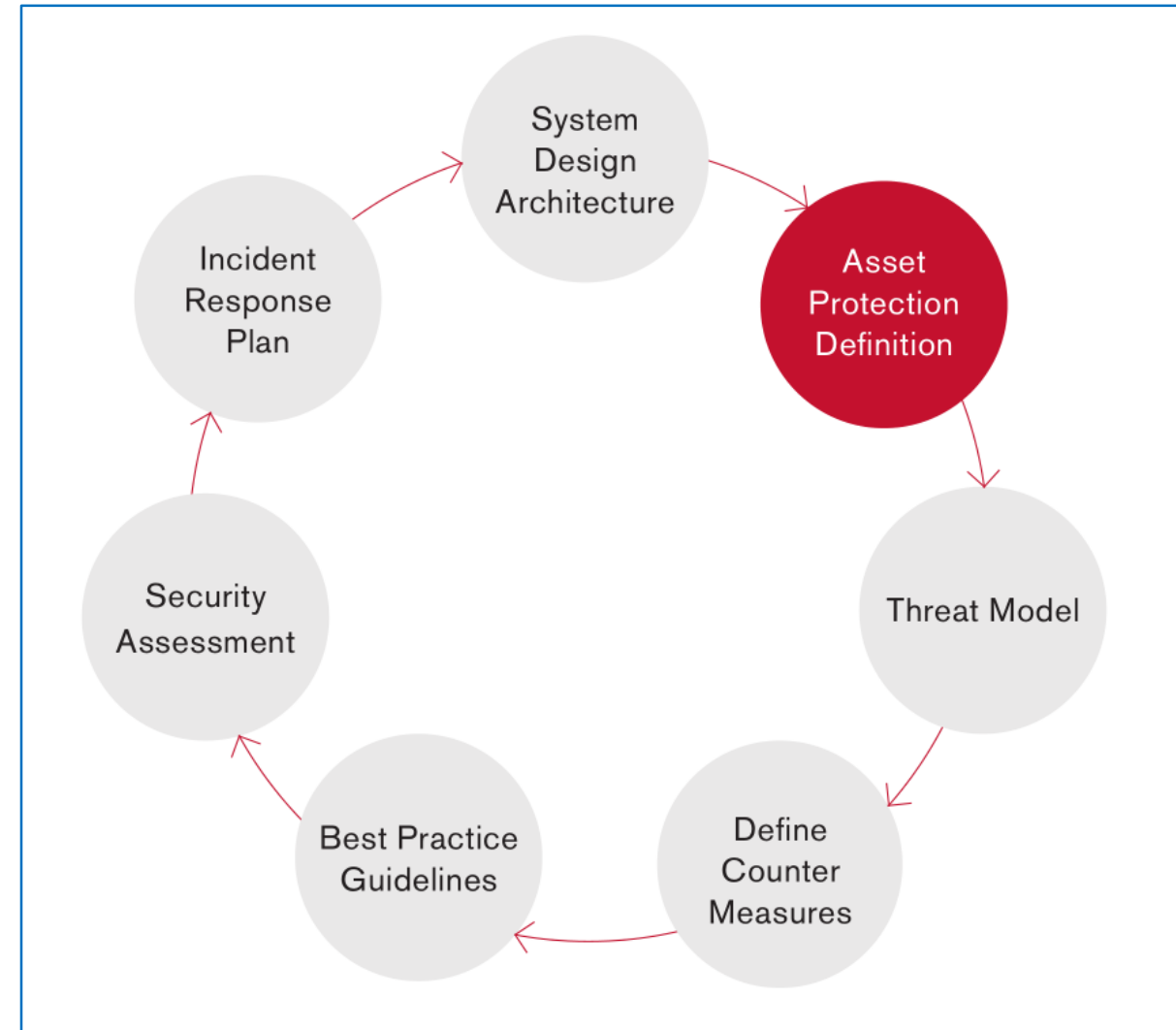
Depending on the functionality associated with a system and the data that it processes, different levels of security assurance will be required. This also means different security controls will need to be implemented. For example, in a cyber-physical system, such as the module that controls braking or steering, a denial of service attack would be a key concern, as it which could affect safety. However, with an in-car app that processes personally identifiable information (PII) or credit card details, confidentiality would be a higher priority.

Other areas of focus may be:

- Anti-theft
- Revenue streams, such as protection of software-enabled functionality
- Intellectual property

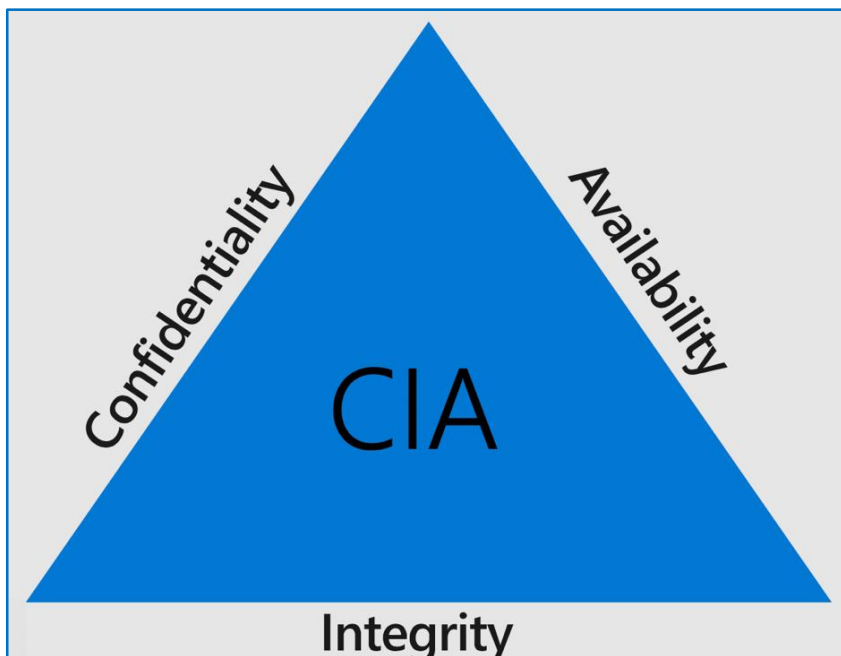
Services within this stage can include:

- Design assessments, including developer interviews
- **Developer awareness training**

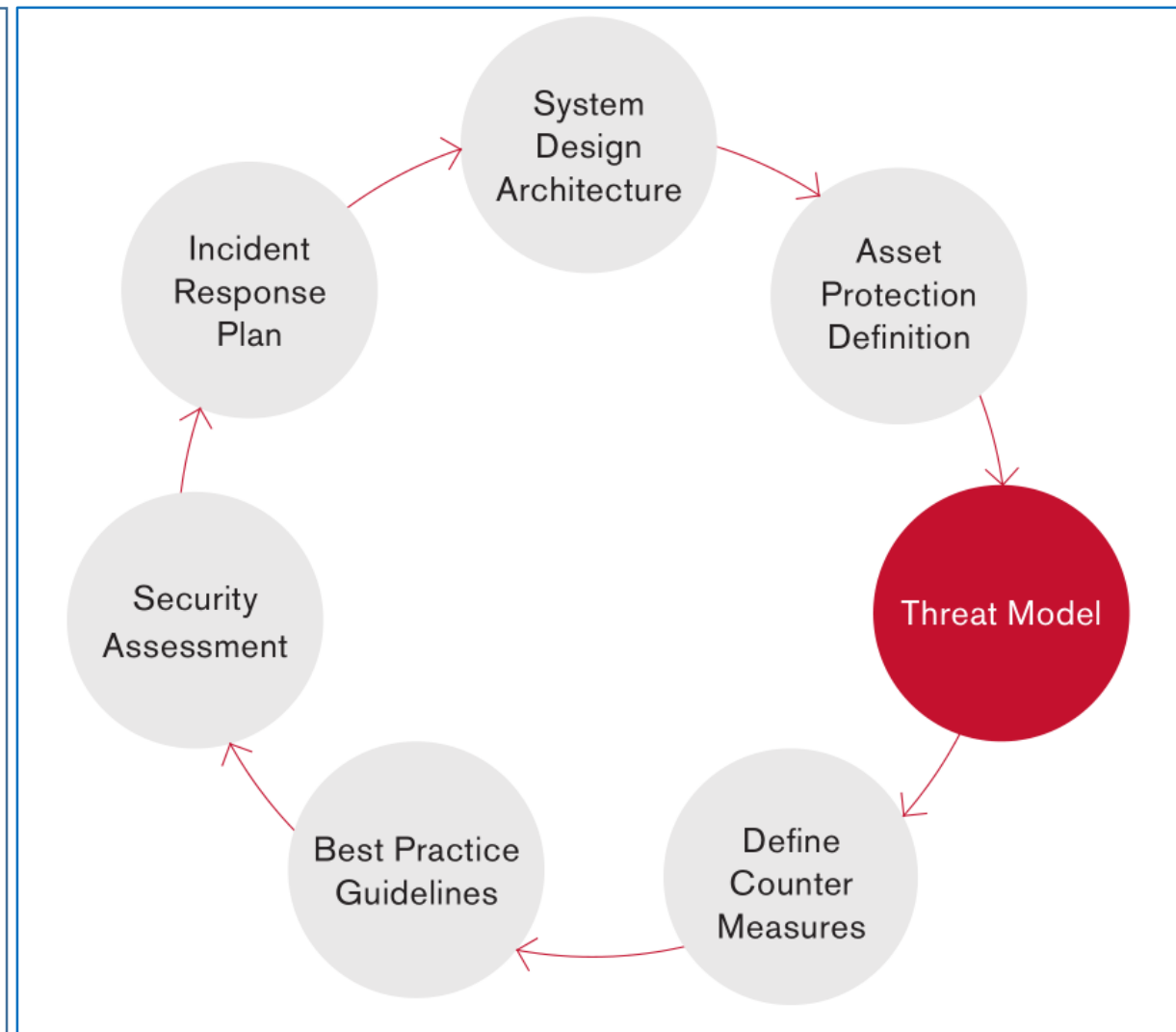


ASDL – Threat Model (mô hình hóa các mối đe dọa)

- Spoofing Identity (giả mạo định danh)
- Tampering with Data (giả mạo dữ liệu)
- Repudiation (thoái thác hành động/trách nhiệm)
- Information Disclosure (tiết lộ/rò rỉ thông tin)
- Denial of Service (từ chối dịch vụ)
- Elevation of Privilege (leo thang đặc quyền)



Tính bí mật – tính toàn vẹn – tính sẵn sàng



ASDL - Define Countermeasures

Many potential attacks identified during threat modelling can be mitigated through **attack surface reduction (ASR) techniques**. Others may need the implementation of security controls such as encryption or authentication.

An important element of ASR is understanding the trust levels required to access each entry point. For each entry point, it is important to consider the importance of the feature that it enables. For **features that are not important to a vast majority of the users**, the feature **should be turned off, disabled by default, or not even installed by default**; users that really want or need it should be forced to take explicit action to obtain that feature.

As a result, any vulnerabilities related to that feature will affect a very small percentage of the product's user base.

We must also consider which classes of users need that feature, and restrict its use to those classes. For example, by default, the feature should not be remotely accessible, allow anonymous access, run with more privilege than is needed, and so on.

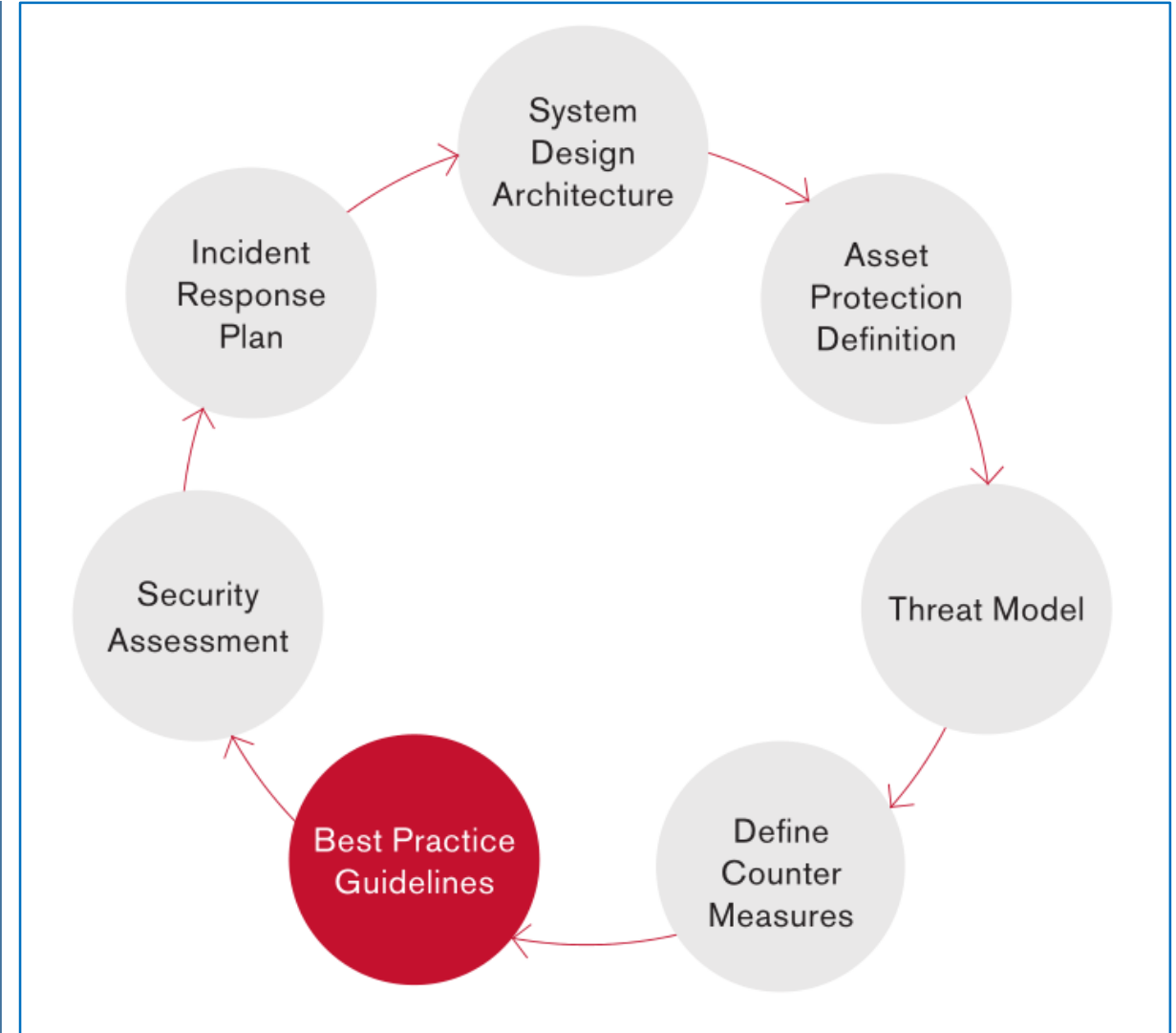


ASDL - Best Practice Guidance

- Implementation guidance on security in resource-constrained systems (Ex: embedded system)
- Technology or service selection security and privacy advice
- Security awareness training around the methods used by real-world attackers

Refer:

<https://developer.android.com/guide/topics/security/cryptography#java>



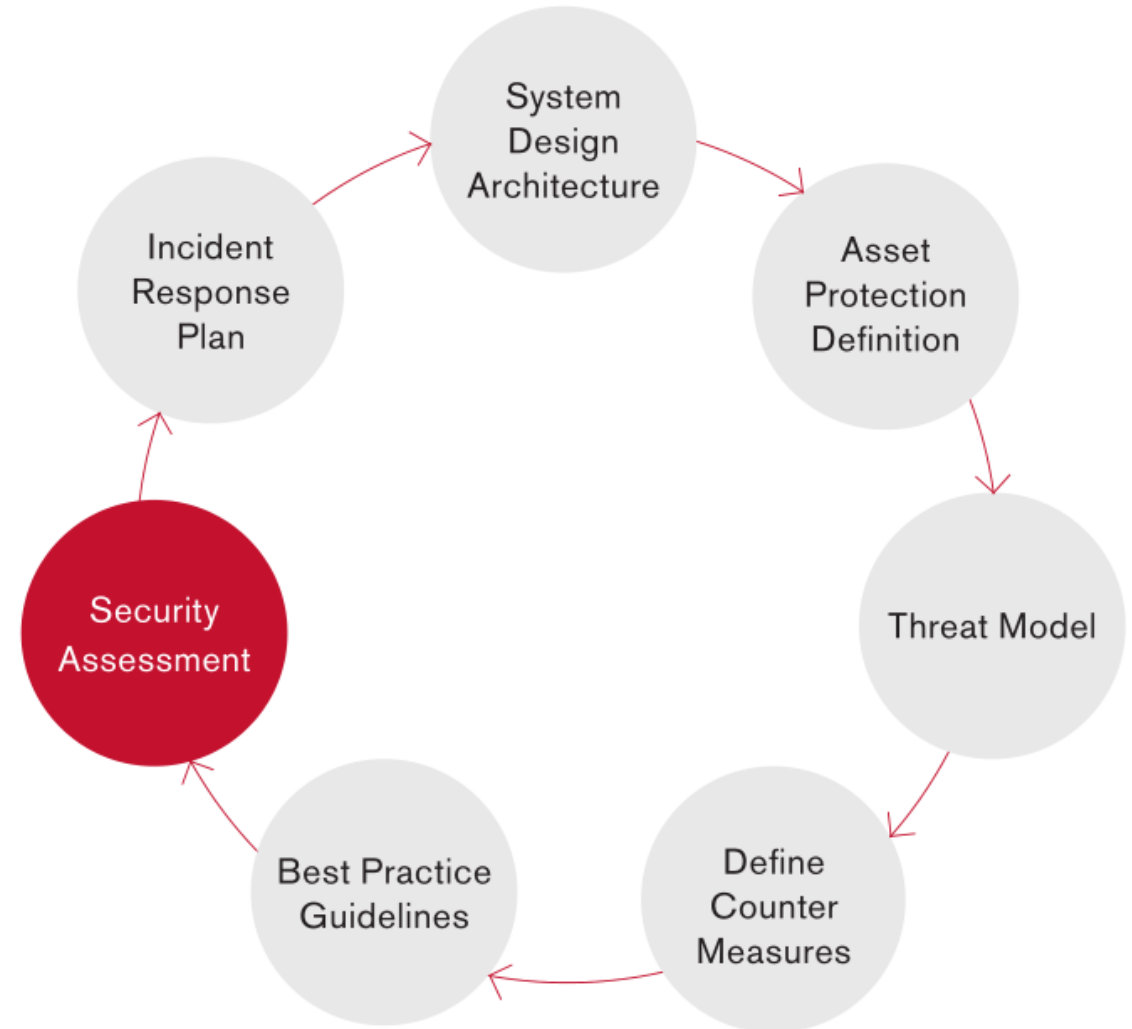
ASDL - Security Assessment

- Penetration and attack testing
- Runtime analysis and fuzz testing
- Static source code analysis

(*)Fuzzing or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program. Ex: Unit test (Vector Cast, Google unit test,...)

(*) Penetration test or pen test, refer: ICAS3.1 pen test-0804.pdf (19 pages)

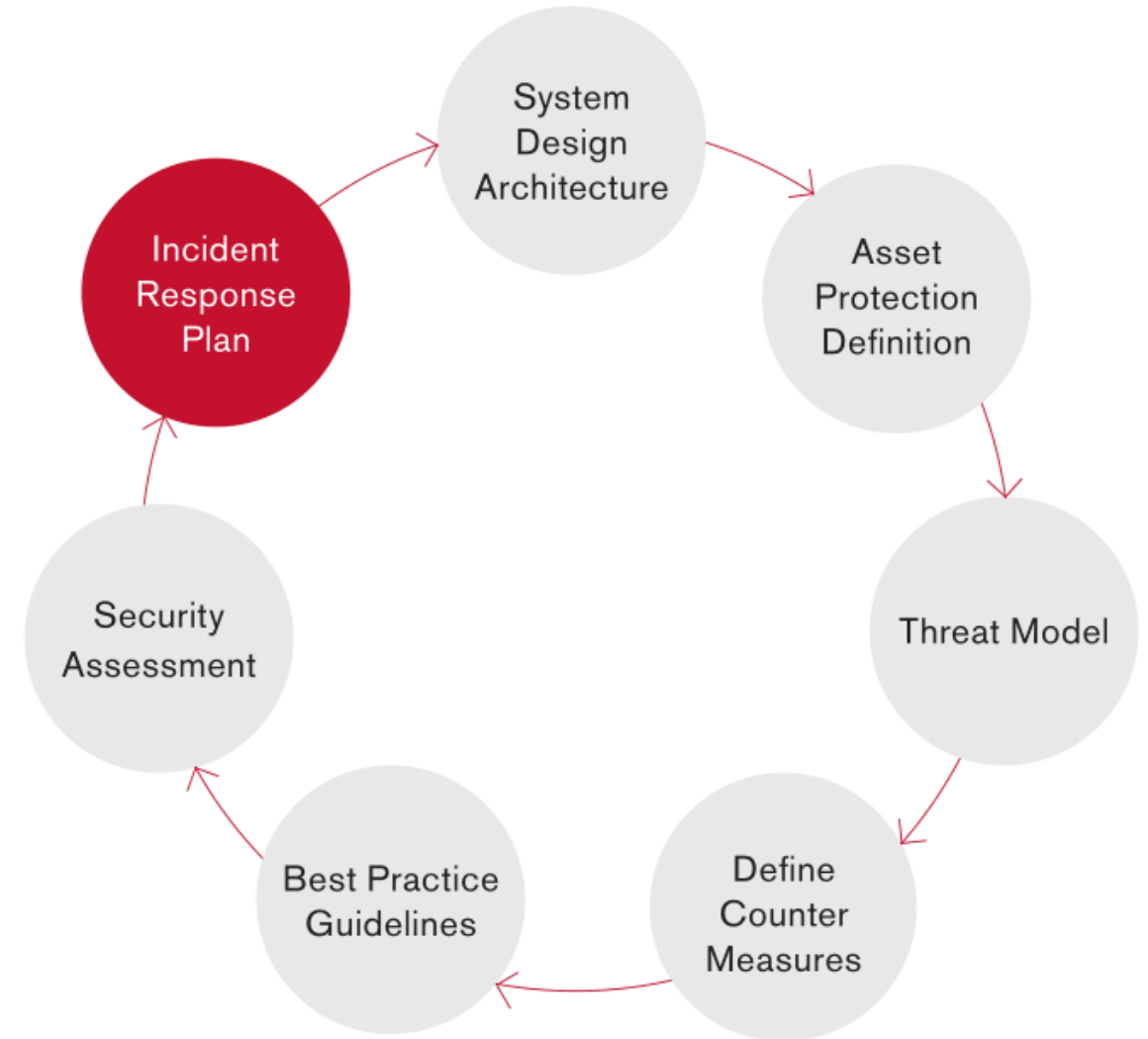
(*) For check code automated, LGE have Misra, Coverity system, beside that have another tool such as Valgrind



ASDL - Incident Response Plan

Security vulnerabilities can still exist within the most rigorously designed and tested systems, so organisations need to develop a plan that will be enacted when an incident occurs. Incidents may take various forms, for example:

- A hacker publicly releases details of a security flaw in one of your vehicle systems.
- Malware is created by a hacker and released in the wild. It exploits a vulnerability that is likely to be present in one of your systems, but you have not yet received any reports that your vehicles have been infected.
- There are reports on the Internet that a hacker has managed to install custom firmware on your telematics module, and they are planning on releasing the details soon so others can do the same.



Integration/System Verification for Security Feature

❖ Integration Security Feature

- Refer: System Architecture Design (SAD), System Requirements, ...
That requirement will have explicit security requirement, such as: use SHA256 (not use MD5) for all hash function, use AES-256 for encrypt/decrypt data, use ECDSA (Elliptic Curve Digital Signature Algorithm) for verify signature/certificate (not use RSA algorithm),...

❖ System Verification for Security Feature

- Static code verify
- Runtime verify
- Feature test (Ex: in Wifi module, tester try change several security option such as: WPA2-PSK, WPA, TKIP,... for try connect to HU AP)
- Use tcpdump, wireshark,... for check packet raw data.
- Use cat, grep,... all config file for try find a “pre know password” of system, if we can find it mean that password is store in cleartext.

Vulnerability Assessment

❖ Same as “ASDL - Security Assessment”

- Penetration and attack testing
- Runtime analysis and fuzz testing
- Static source code analysis

(*)Fuzzing or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program. Ex: Unit test (Vector Cast, Google unit test,...)

(*) Penetration test or pen test, refer: ICAS3.1 pen test-0804.pdf (19 pages)

(*) For check code automated, LGE have Misra, Coverity system, beside that have another tool such as Valgrind

(*)Penetration test almost do check/scan for all know exploit (CVE id), it don't help find new exploit exist in your system.

(*)Attack testing: need more effort for attack to system, try to detect new/others exploit exist in your system.

Refer:

<https://genk.vn/google-treo-thuong-15-trieu-usd-cho-bat-ky-ai-co-the-hack-duoc-con-chip-bao-mat-titan-m-tren-smartphone-pixel-20191122094638519.chn>

<https://genk.vn/internet/cong-ty-israel-vua-hack-iphone-giup-fbi-khong-can-toi-apple-trinh-do-chung-toi-khong-co-doi-thu-20160329113223727.chn>

<http://antoanthongtin.gov.vn/lo-hong-attt/lo-hong-bao-mat-heartbleed-va-nhung-hau-qua-nghiem-trong-100878>

PT_NA1_4007

IVI Penetration Test Report (For A SUVe)



PT-ICAS-011: Store Wi-Fi Password in Clear Text

Level of Risk

Low

Testing Procedure

Log in to ICAS, and there are shm_wlan_ap5g, shm_wlan_ap2g, wlap2.conf and wlap5.conf under the following directories:

```
root@euto-v9:~# grep -r "RHS35" /
grep: /dev/usb-ffs/eap/ep0: File descriptor in bad state
Binary file /dev/shm/shm_wlan_ap5g matches
Binary file /dev/shm/shm_wlan_ap2g matches
Binary file /dev/shm/shm_wlan_gem_req matches
Binary file /data/lge/wlan/wlmb.conf matches
Binary file /data/lge/wlan/wlap5.conf matches
Binary file /data/lge/wlan/wlap2.conf matches
Binary file /home/lge/wlan/wlmb.conf matches
Binary file /home/lge/wlan/wlap5.conf matches
Binary file /home/lge/wlan/wlap2.conf matches
```

The name and password of the hotspot shared by the ICAS are saved in these files:

```
root@euto-v9:~# grep -r "RHS35" /
grep: /dev/usb-ffs/eap/ep0: File descriptor in bad state
Binary file /dev/shm/shm_wlan_ap5g matches
Binary file /dev/shm/shm_wlan_ap2g matches
Binary file /dev/shm/shm_wlan_gem_req matches
Binary file /data/lge/wlan/wlmb.conf matches
Binary file /data/lge/wlan/wlap5.conf matches
Binary file /data/lge/wlan/wlap2.conf matches
Binary file /home/lge/wlan/wlmb.conf matches
Binary file /home/lge/wlan/wlap5.conf matches
Binary file /home/lge/wlan/wlap2.conf matches
```

Description of Risk

The system clearly stores the shared hotspot password and the connected Wi-Fi password which might cause leakage.

Proposed Fix

The passwords of shared hotspots and connected Wi-Fi passwords are stored in encrypted form.

Application in DCV project

- ❖ We should have more carefully for these attention below:
- Which module have interact with user sensitive data ?
=> Bluetooth (message, contact, audio), Wifi, Phone Call,...
- Which module have connect with other device/network ?
=> Bluetooth (message, contact, audio), Wifi, Android Auto, Apple Carplay, Log service,...
- Which module/program have function execute command ? Such as MCU/ECU command, shell script, Diag setting,... ?
=> Secure shell (SSH on port 22), ADB shell, Linux shell (/bin/sh, /bin/bash), Python, TestIF (work with Ignition tool, ODIS), ...
- Which module/program can connect to internet ?
=> Online service, ...

Security standards (framework, protocol/mechanism, algorithm,...)

❖ Refer: CBL-Introduce-about-cryptography.pptx

\\dcv-data\Video-Training\01. SD\02.Technical\01.Programming Languages\01.C++\Overview using hash - crypto system in project - By Nguyen Van Luan